REMARKS/ARGUMENTS

Claims 1–23 remain in the application for further prosecution.

Claim Rejections - 35 U.S.C. § 103

Claims 1-7, 11-14, 16-19, 22 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application Publication 2002/0049909 ("Jackson") in view of U.S. Patent No. 5,644,704 ("Pease") and U.S. Patent No. 7,149,801 ("Burrows").

Claims 8-10, 15, 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jackson in view of Pease and Burrows, further in view of U.S. Patent No. 6,842,860 ("Branstad").

As explained in previous responses, the present claims are directed toward more rapid authentication in gaming machines. Due to the need to comply with gaming regulations as well as the desire to insure that wagering games are not tampered with, gaming software is authenticated on a periodic basis and on the occurrence of certain events such as if a door is open in the cabinet, on reset or when the wagering game commences. Secure hashing is used to produce a unique representation of the software that must be authenticated. In order to produce a unique hash, complex computational steps must be performed on the underlying data. The need for secure hashing for authentication is balanced by the desire to rapidly authenticate game software to minimize delays in playing the game. The present claims address both these concerns by calculating hash key-values from non-sequential data blocks based on random starting addresses. In this manner, the speed of authentication is enhanced because not every data block in a media is used in determining the hash key-value. However, security is insured because a sufficient amount of the data is verified by a unique hash key-value to meet security requirements.

Claim 1 Is Allowable Over The Impermissibly Combined References

The Final Office Action combines three separate references in the obviousness rejection against claim 1. Applicant respectfully submits that the combination of these references to achieve the claimed subject matter constitutes impermissible hindsight. The Final Office Action initially cites Jackson, which discloses a basic authentication routine for the software of a gaming device. Because Jackson requires authentication of all of the software on a media device, it suffers from the problems explained above and is an example of known authentication art. Claim 1 departs from the teaching of such art to achieve certain advantages noted above. The Final Office Action then applies two disparate references in combination with Jackson to stitch together the remaining elements of claim 1.

Jackson is representative of commonly known authentication art in the gaming industry. Jackson relates to a secure authentication routine for gaming devices that authenticates every data block of a gaming software program during program operation. (¶ 81). Jackson's authentication method, while meeting security requirements, takes time, especially for larger game software programs, and therefore slows down wagering game operation. The Final Office Action acknowledges that Jackson does not teach "setting an address pointer ADDR to a first next memory location in the device, setting the next ADDR to a next memory location in the device to be authenticated, and adding a predetermined number N to the ADDR such that a next ADDR = ADDR + N, and that N is equal to a positive or negative integer excluding -1, 0, and 1." (pp. 3-4).

The Final Office Action cites Pease for setting an address pointer to a first memory location and setting a next pointer to the next memory location. Pease relates to a general method of authentication that uses unused memory 27 for part of the random-looking (but

reproducible) data (termed non-sequential data) at boot time in the authentication calculations. (Fig. 2). Pease uses a non-associative technique in a circular way to verify the data, but does not always start at the beginning of the data blocks in the device in the verification process. (Col. 4, ll. 20-38). Pease also discloses authenticating every file in the device, thus resulting in a process that has adequate security but as with prior art such as Jackson requires excessive time and processor resources to authenticate every block in a file. (Col. 4, ll. 33-38).

The Final Office Action combines a third reference, Burrows, to Jackson and Pease. Burrows relates generally to designing a "puzzle" or problem that is easy to generate and hard to solve for a machine. (Col. 1, ll. 17-21). This is used for spam prevention, as a sender of a spam message will not devote the resources to solving a problem generated by Burrow's functions but a sender of limited e-mails will have the necessary computer resources to solve the problem to send the e-mail, thus providing an indication that an e-mail message sent by the computer is not spam. (Col. 1, ll. 14-18). Burrows is directed toward the deficiencies of present spam detection systems because different computers present the "puzzle" at different speeds resulting in undesirable delays for slower computers. (Col. 4, l. 63 to Col. 5, l. 3). Burrows relies on the fact that computer memory speeds (i.e. RAM memory speeds) are far less variable than CPU speeds. Burrows creates a puzzle that can be solved in reasonable time by those systems that have a minimum amount of memory (larger than the largest processor cache) and therefore allowing such systems to send non-spam e-mail messages. (Col. 5, ll. 14-19).

The section cited by the Final Office Action uses a checksum to rapidly match a received solution (X_0) to the "puzzle" to the acceptable solutions stored. If the solution is stored, indicating a correct solution to the "puzzle," the message may be sent. (Col. 12, ll. 34-38). Without the checksum, the puzzles generated may have multiple solution sets. (Col. 12, ll. 29-

34). The checksum is used to disambiguate the multiple solution sets to arrive at the single solution that is expected as the solution of the "puzzle." The checksum is calculated over a short sequence of values, which can be a subset of the sequence forming the puzzle solution. A subset of the path may be chosen because the checksum of the subset is sufficient to find the correct puzzle solution as there are only a few solutions within the solution sets. (Col. 12, ll. 38-40).

The check sum disclosed in Burrows is therefore directed to help search for an item (acceptable solution to a "puzzle") and is not applicable to the security field. It is not necessary to calculate a check sum for all the solutions because it is unnecessary to determine an exact match. The checksum is designed to help narrow the correct solution to send the e-mail i.e. a solution to the "puzzle." The checksum in Burrows is used to find the same solution as that received by the e-mail server to disambiguate. However, anybody can fake the checksum value in Burrows because it is not complex and not designed for a relatively unique representation of the solution data.

In contrast, claim 1 requires a hash algorithm which is a secure function that takes an input string (usually a string of digits) and converts ("hashes") it to a fixed size, usually smaller, output string ("hash value"). The object is to "fingerprint" the input string so that the resulting hash value is very likely to represent one and only one input string. A hash has several properties that allow its application to secure authentication. For example, changing a single bit of the input string will result in a different hash value thus indicating tampering of the data.

The hash is also a non-invertible function that prevents duplication of a unique hash for message. An invertible function such as a checksum allows the same value for different data. For example, if two blocks were switched, the checksum is the same, but the hash value would be different. This property is necessary for security because alterations in the underlying data

such as changing the order of blocks or changing the underlying data may be detected in an alteration of the hash value. In fact, the secure properties of hashing algorithms allow the compliance with security-based governmental regulations for wagering game machines.

As simple mathematical functions such as checksums are easily subverted, they are not appropriate for effective security. For example, a data set may be easily modified to produce the same checksum as a different data set. One of ordinary skill in the art would recognize that "these types of redundancy check are useful in detecting *accidental* modification such as corruption to stored data or errors in a communication channel. However, they provide no security against a malicious agent as their simple mathematical structure makes them trivial to circumvent." (http://en.wikipedia.org/wiki/Checksum). A checksum is thus insecure and could not be used for reliable verification of a unique set of data. Checksums therefore do not meet governmental regulations relating to wagering game machines. The applications and requirements for a secure hash such as the SHA-1 algorithm in the authentication field would preclude checksums as taught by Burrows because checksums are not even remotely secure.

A person of ordinary skill in the art would not reference Burrows if it were desired to make an authentication process more efficient. First, Burrows does not deal with authentication (i.e. a proof of origin) or security issues and therefore is in different field of technology than that of the claims (or Jackson and Pease for that matter). The checksum in Burrows is really a shortened identifier such as a first name, used to distinguish the requested solution from a set of possible ones. Contrary to the assertion of the Final Office Action, one of ordinary skill in the art would not look to apply a checksum of a tree method for locating an object for the purpose of spam prevention in Burrows to an authentication problem as in the present claims. The use of a checksum for a relatively discrete set of data is an inherently insecure method and does not meet

the requirements for security which mandate the use of a secure hash for unique representations of a data set.

Second, one of ordinary skill in the computer security field would not look to solutions relating to checksums because they are inherently insecure. As explained above, checksums are easily replicated for different sets of data thus providing no security function whatsoever. Particularly, in view of gaming applications, one or ordinary skill would not reference Burrows because a check sum could not meet applicable regulations governing wagering game machines. The combination of Burrows with references requiring secure hashing such as Jackson or Pease is therefore improper and claim 1 is non-obvious over the cited references.

Claims 13 and 17 Are Allowable Over The Impermissibly Combined References

The Final Office Action uses similar rationales to reject claims 13 and 17 based on Jackson in combination with Pease and Burrows. The Examiner concedes that Jackson "does not explicitly disclose memory locations in the form of addresses or the like, that the hash calculation is performed on a sample of memory locations being a number of memory locations that is less than all of the plurality of memory locations and that each memory location of the sample of memory locations is separated from other memory locations of the sample of memory locations by at least one memory location." (p. 8). The Final Office Action concludes that one of ordinary skill would incorporate the checksum techniques of Burrows into Jackson in order to increase the speed of the integrity checks. (pp. 9 and 11).

As explained above, one of ordinary skill in the art would not apply the techniques of Burrows to Jackson or any other security application because the references are in different fields. Moreover, because of the unique requirements of secure authentication and verification, one of ordinary skill would not look to the field of searching to apply checksum techniques to a

security method such as Jackson or Pease. Claims 13 and 17 are therefore allowable because the combination of Burrows with Jackson and Pease is improper.

Conclusion

It is Applicant's belief that all of the claims are now in condition for allowance and actions towards that effect is respectfully requested.

If there are any matters which may be resolved or clarified through a telephone interview, the Examiner is respectfully requested to contact the undersigned attorney at the number indicated.

Respectfully submitted,

Date: May 8, 2008

Wayne L. Tang
Reg. No. 36,028
NIXON PEABODY, LLP.
161 N. Clark Street, 48th Floor
Chicago, Illinois 60601-3213
(312) 425-3900
Attorney for Applicants